NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ANALYSIS, DESIGN, AND IMPLEMENTATION OF A
DATABASE MANAGEMENT SYSTEM FOR THE NAVAL
POSTGRADUATE SCHOOL ALUMNI ASSOCIATION

by

Mark T. Kohlheim

September 1992

Thesis Advisor   Shu Sheng Liao

Analysis, Design, and Implementation of a Database Management System for the
Naval Postgraduate School Alumni Association

by

Mark Todd Kohlheim
Lieutenant, United States Navy
B.S., Oregon State University, 1982

Submitted in partial fulfillment of the requirements for
the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September 1992

ABSTRACT

The Naval Postgraduate School Alumni Association is a professional group consisting of graduates from the Postgraduate School. The Alumni Association is a self supporting entity that is not part of the administrative structure of the school. The Alumni Association supports itself through membership fees paid by graduates who join the association. The administrative work required to run the association is accomplished through a part-time employee and volunteers.

Until recently, the Alumni Association was unorganized and lacked a firm plan of action to establish itself with the School's graduates and to increase its membership base. Records of graduates were located in various locations and in different formats which made tracking and contacting alumni an extremely time consuming operation. This thesis develops a database management system for the Naval Postgraduate School Alumni Association. This system provides a standardized format for storing data and tracking alumni. It also performs the time consuming accounting and billing functions associated with the Association's membership management. This thesis provides an outline covering the Alumni Association's system requirements analysis and design methodology. The system was written using dBASE IV, version 1.1.

TABLE OF CONTENTS

I. INTRODUCTION

A. BACKGROUND

The Naval Postgraduate School Alumni Association has been in existence for a number of years but has not been able to become a viable entity. The Association's goal is to provide a professional network for the graduates of the Naval Postgraduate School. The Alumni Association is independent of the school in funding and administrative support. The Association administration is run through the efforts of volunteers and part-time hires.

The inability of the association to become a viable entity can be attributed to a few major factors. First, the fluctuation in the work force effort and the lack of consistent commitment to the Association have caused inefficient and misguided efforts. The second major factor effecting the viability of the Alumni Association is the lack of an accurate and homogeneous database of the school's alumni. Alumni records over the history of the school have been maintained in a variety of methods. Once the student had graduated the records were archived, thus making access to these records difficult and time consuming.

The new Alumni Association President decided that the Association should use the resources available to it and have a computerized database management system developed to handle the data requirements of the Association. The basic requirements for the system included the storing of Alumni data for quick and efficient retrieval, which would also provide mechanisms for the updating of the stored data. Additionally, the system would be required to automate and perform the routine accounting and billing functions necessary to maintain the membership records of the Association. This thesis accomplishes these requirements and

includes an in depth overview of the software application's analysis, design, and implementation.

B. ALUMNI DATABASE MANAGEMENT SYSTEM (ADMS)

The Alumni Database Management System (ADMS) was developed to make the management of Alumni data easier and more efficient, as well as automating the business functions of the Association's membership accounting. To accomplish these objectives, it was necessary to analyze the data requirements necessary to maintain the Alumni Association's database. This also required the identification of the business processes used in the maintenance of the Association's memberships so that they could be automated.

Ashton-Tate's dBASE IV, version 1.1 was used to develop this system. Using dBASE IV's application generator, a series of application prototypes were developed using requirements input from the Alumni Association's management. The final system is the result of the refinement of the user's needs through iterations of the prototype process.

ADMS is a menu driven system that provides a "user-friendly" interface to the application. The system provides the mechanisms necessary to manage the Alumni database, generate periodic reports, and perform the monthly billing processes, as outlined in Appendix D. The Alumni Database Management System is also designed so that the user will be able to generate one-time queries and reports with minimal effort as well as expanding the system as future needs warrant.

C. CHAPTER DESCRIPTION

Chapter II will discuss the Definition Phase of the applications development. This chapter will define the scope and objectives of the application as well as how these objectives will be accomplished.

2

Chapter III will discuss the application development methodology used in developing the Alumni Database Management System. The definition, requirements, evaluation, design, and implementation phases will be discussed.

Chapter IV, Conclusions, will discuss the functionality of the system and possible areas for further development in the future.

Appendices A through D provide supporting documentation on requirements, data dictionary, application documentation, and a user's guide.

II. DATABASE APPLICATION DEVELOPMENT

The Alumni Database Management System was developed using five phases. These phases will be discussed in this section. The methodology used for each phase will be discussed followed by the application of that methodology in the development of the ADMS.

A. PHASE I: DEFINITION PHASE

1. Methodology

The definition phase is the simplest phase of the project development phases. The goal of this phase is to determine what has to be accomplished. The scope of the project has to be established as well as the project's feasibility, which includes cost, technical, and schedule considerations.

2. ADMS Definition

The goal of this project was to develop a database management system for the Naval Postgraduate School Alumni Association. The scope of the project met the requirements for a thesis project. The cost of the project was not considered to be an issue because the hardware and software were available from the Alumni Association. The work was completed on a 286, 20 MHz IBM compatible PC. The schedule was set to begin in January 1992 and system completion date set for August 1992. The time allotted for the project was considered adequate to complete the project.

The system's functionality objectives were established during numerous interviews with the Alumni Association's President, Captain Ferrell. The final functionality objectives were agreed to by Professor Liao. Phase I was accomplished in approximately one week.

B. PHASE II: REQUIREMENTS PHASE

1. Methodology

The goal of this phase is the attainment of very specific system objectives. These specific system objectives are the result of determining the user's requirements. It is crucial that the system developer knows exactly what the system must do before it can proceed to the development phase. Well defined and understood user requirements are vital to ensure that the final system does what the user intended for it to do. ·

The first step in this process is to define the database objects. A database object is a grouping of properties that can be used to describe an entity. An instance of an object is a unique occurrence of that object. The second step in the process is to determine the functions and operations that the application will perform using the database. The most effective method of performing these steps is by conducting user interviews. Normally a series of interviews will be required. Examining existing reports and other outputs of current procedures will also be useful in determining user requirements. Building prototypes based on user interviews can be used to verify user requirements and help define additional requirements.

2. ADMS Requirements

Interviews commenced with Captain Ferrell the second week of January 1992. Captain Ferrell had a general idea of what he wanted the system to do. He was not going to be the primary user of the system, but the system needed to support his needs. The initial interviews lasted for approximately two hours.

After the initial interview some sample reports and screens were developed. These reports and screen designs were presented to Captain Ferrell during a second interview so that the requirements of the first interview could be

verified as well as identifying any additional requirements. Based on this information a prototype was developed based on the sample data entry screens and reports. This was presented late in January 1992. Minor changes were required and no further prototypes were necessary.

a. Data Requirements

Base on the user interviews and prototype process, it was determined that only one object was necessary to meet the requirements of the system. The ALUMNI object is shown in the Object Diagram, Table 1, Appendix A. The use of only one object (non-relational) would also meet the user's requirement that the interaction between the user and the database be as simple as possible to facilitate one-time queries and report generation. Appendix A, Table 1 provides the object diagram of the ALUMNI object. Each item listed in the object represents characteristics of the Alumni that need to be used by the system. Appendix A, Table 2 provides the object definition, which lists all the object's properties and each property's domain. Appendix A, Table 3 provides the domain definitions that specify the format of each domain. This information will be used in Phase IV during the database design.

b. Application Functional Requirements

To maintain the records of the Alumni Association, a clerk would be required to perform recurring administrative functions. These functions include record entry, display, editing, deletion, membership accounting, and report generation. The data flow diagram (DFD) shown in Figure 3.1, shows a graphical model of the administrative processes that need to be performed to run the Alumni Association. This diagram will serve as an aid in the design of the Alumni Database Management System.

6
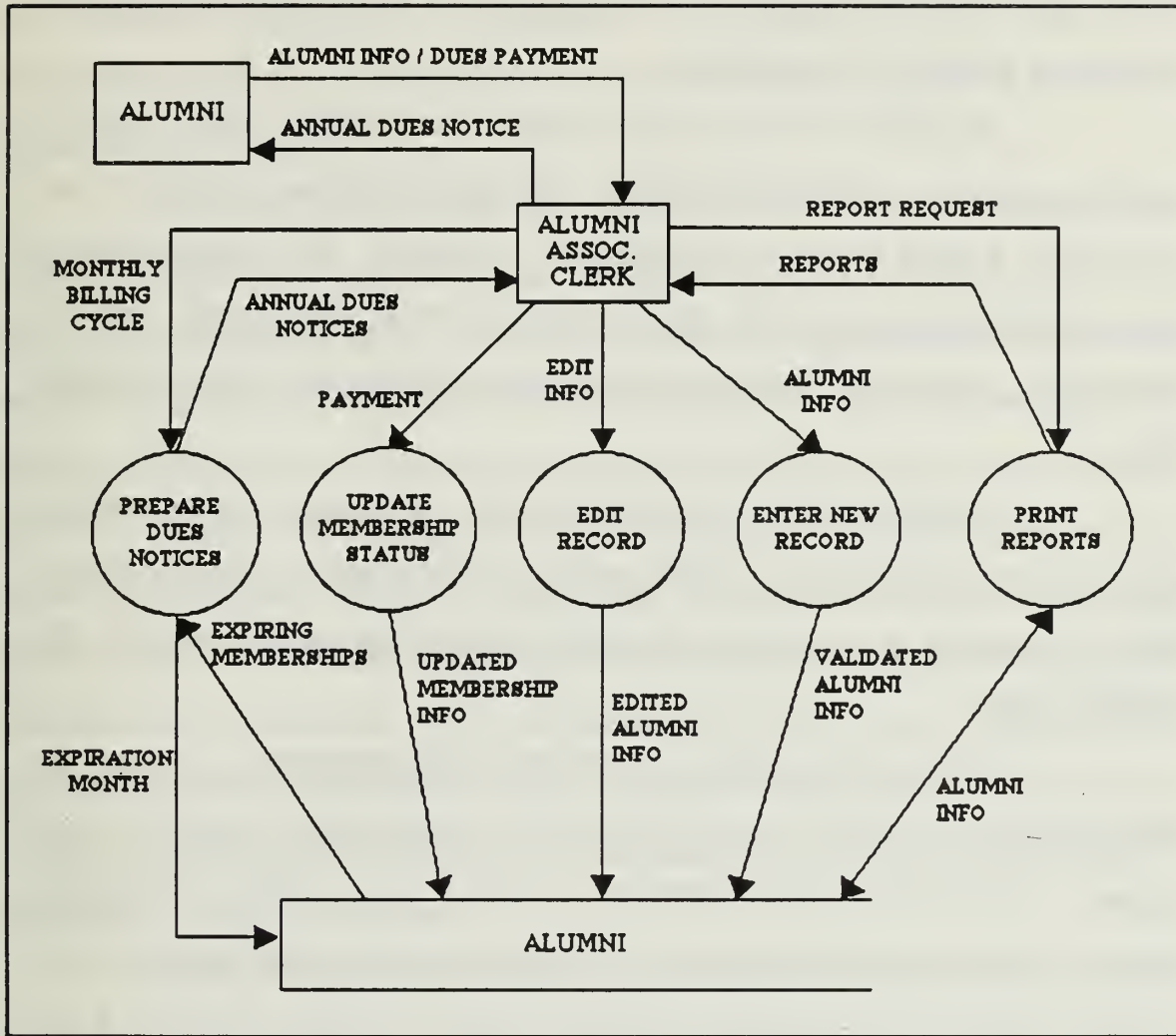
Figure 3.1   Context Data Flow Diagram

The DFD consists of four elements. They include: data flows, which are represented by an arrow; processes, which are represented by circles; data sources/sinks, which are represented by squares; and the data store, which is represented by an open ended rectangle.

The Alumni Association clerk will enter new Alumni data into the system. The Alumni's last name, first name, and middle initial will be used as a unique identifier for each record in the system. As shown in Figure 3.1, the

ENTER NEW RECORD process requires the input of an Alumni's data and will create a new instance in the data file.

The EDIT RECORD process requires the entry of a valid Alumni's name that is currently stored in the data file. The process searches the data file and retrieves the Alumni record for modification or deletion. The record, with its changes, will then be store in the data file replacing the original record. Table 4, Appendix A, summarizes the update mechanism requirements for the ALUMNI object.

The PRINT REPORTS process, upon user request, will retrieve the relevant report data from the ALUMNI data file and generate the requested report. Table 5, Appendix A, summarizes the display mechanism requirements for the ALUMNI object.

The last two processes, PREPARE DUES NOTICES and UPDATE MEMBERSHIP STATUS, require the user to initiate these functions when required. PREPARE DUES NOTICES is similar to the PRINT REPORTS process. It will retrieve the appropriate data for memberships that are about to or that have expired and it will generate notices. Again, see Table 5, Appendix A, for the display mechanism requirements for the ALUMNI object. The UPDATE MEMBERSHIP STATUS process is used to update the membership status of selected records. This process is similar to the EDIT RECORD process except for the retrieval and modification of specific data elements.

C. PHASE III: EVALUATION

1. Methodology

This stage of the system's development evaluates the information gathered during the Requirements Phase to determine if the project is still feasible with respect to cost, technology, scope and schedule. If the system's development

is considered feasible, then the evaluation of the different system architecture alternatives need to be evaluated so that the best match is made for the user's present and future needs.

Evaluating the cost, technology, scope, and schedule of the project at this point is extremely important so that the project will not continue without an achievable finished product. Cost is an important consideration because the system's requirements can grow beyond the available budget or the worth of the project to the company. Technology has to be considered to ensure that the system does not require hardware or software features and performance that are not currently available. In addition, leading edge technology requires that the added risks and costs be evaluated. Scope and schedule need to be reevaluated using the specific user requirements. This will ensure that the size of the project has not grown during the Requirements Phase. Even if the scope of the project has not changed, the project's schedule should be reevaluated to ensure that enough time is available to complete the project.

The final step of the evaluation process is the selection to the best system architecture for the system. This step can only be undertaken when the system's requirements fit into the system's development constraints or when a compromise between the to is reached.

2. ADMS Evaluation

Evaluation of the Alumni Database Management System project was accomplished quickly due to the system's architectural constraints. The first constraint was on the system's hardware. The system had to be compatible with the current microcomputer that the Alumni Association was using. This computer was an 286 / 20 MHz IBM compatible. The hardware constraint did not pose any technology barriers to the development of the proposed system. The second

constraint on the system concerned software. The application had to be developed using Ashton-Tate's dBASE IV software package that the Alumni Association owned. A review of this software and its capabilities showed that this software package would meet the current and future needs of the application.

A review of the user's requirements generated during the Requirements Phase showed that the ADMS could be completed and delivered on time. Development of the system would utilize the equipment currently in the possession of the Alumni Association and no additional costs would be necessary. Some recommendations for future development and system enhancements, discussed in the conclusions, will require some additional costs.

D. PHASE IV: DESIGN PHASE

1. Logical Database Design

The logical database design is developed as the structure of the physical database and the applications that it must support. The logical design is developed from the system's objects and their relationships. A system which has many objects will have relationships between them. These objects can range from a simple object (the most basic) to an aggregation object, which represents a group of entities. In addition, a system with many objects will have relationships between them. These relationships can be as simple as a binary (one-to-one) relationship or as complex as a many-to-many relationship. A relational database design is necessary to develop a system to handle these relationships.

The Alumni Database Management System has one object, therefore there are no relationships that need to be designed into the system. Logically there is one relationship in the interaction between the Alumni Association and an Alumni. Membership in the Alumni Association is dependent on being an Alumni from the school, but being a member of the Alumni Association is optional for all

Alumni. This relationship is not designated by the creation of two objects due to the user's requirements that the database structure be made as simple as possible to facilitate simple interaction between the user and the database during one-time queries and reports.

2. Normalization Process

The normalization process is necessary to test the relationships between the objects of the system. This process will prevent the designing of modification anomalies into the database structure. These anomalies, such as update and modification anomalies, will create undesirable effects when working with the actual database. The normalization process is the technique used to eliminate these database modification anomalies.

There are seven normal forms that a relation can take. Each normal form is a progression from the normal form that preceded it. Domain/Key Normal Form is the highest level of normalization that can be obtained. When a relation reaches this level it is guaranteed not to have any anomalies. Relationships may satisfy one or more of the normal forms but may not satisfy them all. The more normal forms that a relation satisfies the less likely that a modification anomaly will occur.

The normalization process is critical for relational database designs because of the relational dependencies between objects. For a non-relational database design only the first few normal forms need to be evaluated.

a. First Normal Form

This normal form requires that the relation has no repeating groups. ALUMNI has no repeating groups, meeting the requirements of the first normal form.

b. Second Normal Form

This normal form requires that all non-key attributes must be dependent on the key. ALUMNI has a compound key (LNAME, FNAME, MI) that uniquely identifies each tuple. ALUMNI is in second normal form.

c. Third Normal Form

This normal form requires that the relation is in second normal form and that it has no transitive dependencies. This normal form does not apply to a non-relational design and the remaining normal forms are also not applicable.

3. Physical Database Design

The physical database design transforms the logical database design into the framework to meet the specific data element structures required by the database management software being used. The software being used for this application, dBASE IV, provides for six types of data elements as well as the naming of these data elements using a name that does not exceed ten characters. The six data types used in dBASE IV are:

1. Character - textual information
2. Numeric - numeric values
3. Float - numbers with varying amounts of decimal places
4. Date - calendar dates stored in the mm/dd/yy format
5. Logical - Boolean expressions (T, F, Y, N)
6. Memo - storing of large amounts of text

The Alumni data file contains 26 data elements, which are displayed in the Data Dictionary shown in Table 6, Appendix B. These data elements provide the necessary data to maintain Alumni records and Alumni Association membership records. The Data Dictionary's data elements represent

approximately a 25 percent decrease in the number of data elements that the Alumni Association was maintaining but were not necessary or used.

4. ADMS Application Design

An application extracts data from the database for the user by utilizing menus, forms, batch processes, programs, and reports. Application design is the last task to be completed for the Implementation Phase.

a. Scope of Application

The functionality of the application was determined during the Requirements Phase. The processes to be performed were designed into modules that could be broken down into the fundamental functions that the application was to perform. The structure and scope of the application are depicted by these modules. With this basic design developed, an application prototype was developed to demonstrate the application to the user. This permitted the user to view the proposed application and discuss any modifications before implementation.

b. Control Mechanism

This step of the design process involves the selection of the method to be used for controlling the application. The two primary methods that can be used and that were considered for this application include a menu driven mechanism and a command driven mechanism. The menu driven mechanism was selected because it provided the most user friendly interface. This mechanism also permits the application's processes to be displayed in a logical sequence.

c. Menu Options

An action/object menu design hierarchy was selected to be used in the design of the application. This type of structure will allow the user to start with a list of general actions that would lead to specific actions to be performed.

13

This would be accomplished through the successive selection of menu options until the desired action is reached. The menu hierarchy is shown in Figure 3.2.
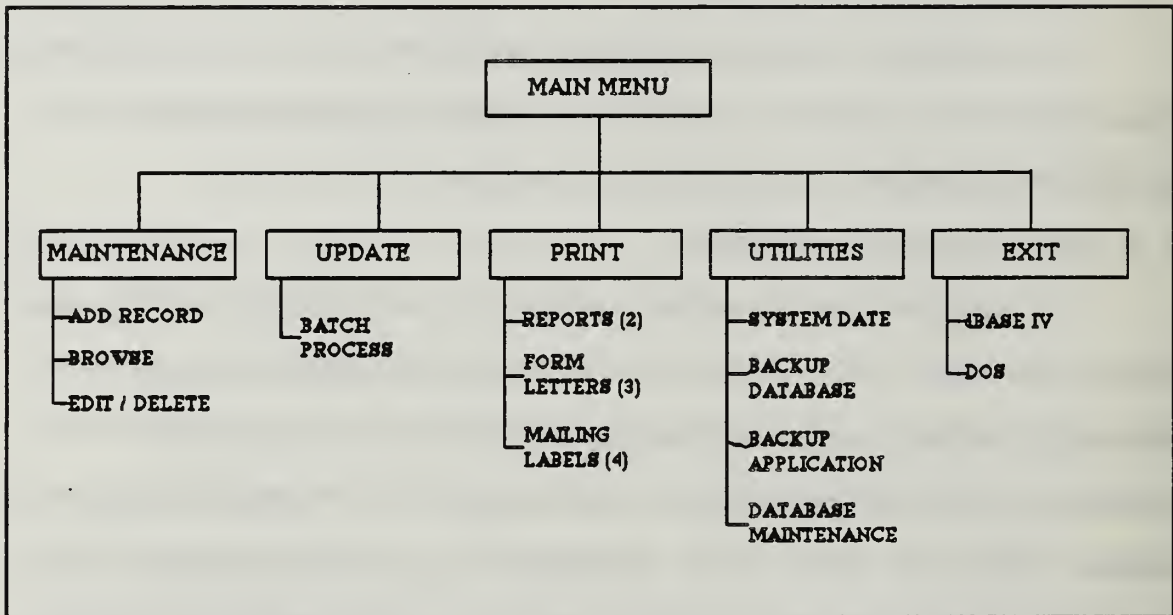
Figure 3.2   Menu Hierarchy

d.   Data Presentation Design

Decisions concerning the format of data presentation were discussed during the requirements interviews held with the Alumni Association President. It was his desire that the reports and form letters, that the system would generate, would be exact duplicates of the ones currently being used. Samples of new reports, which would be incorporated into the application, were produced by hand for approval. All the reports and form letters were then developed, using dBASE IV's report generating features, so that the user could see exactly how the output from the application would look. This allowed the user one last opportunity to incorporate any additional changes for the implementation of the application. Table 7, Appendix B, details each report and its view file.

Screens that were to be used in the application were also designed at this time. This would allow the user to see how the data would be displayed on the computer screen and give the user the opportunity to request any changes to the screen format or layout.

The screens used for adding and editing records are identical in design except that the adding record function will present blank data fields and the edit function will initially display the first record in the data file (files are indexed alphabetically). The data display screens were designed so that the user would be able to enter data quickly and efficiently. The screen was designed so that as the user began to enter data, the flow of data fields was in a logical order so that the data fields were grouped according to type of data (e.g., address data fields were grouped together in a logical sequence). When possible, default values were used in data fields to assist the user in data entry.

The use of color was also incorporated into the design of the application's screens. This enhanced the distinction between data prompts and data fields. It also made the application's screens more esthetically appealing to the user.

E. PHASE V: IMPLEMENTATION

1. System Programming

To ensure success in the construction of the application it is important that the user's requirements be frozen at this point and that the application be built in accordance with the design developed during the Design Phase. dBASE IV provides an application generator that will automatically code most of the functions that the application will perform. The application's customization was accomplished by embedding code before or after the standard dBASE IV code generating capabilities. The applications code is provided in Appendix C.

15

Before the application can be constructed, all the screens, reports, and database must be created. The resources that will be used by the application should all be placed into the same dBASE IV control panel. The dBASE IV application generator will integrate all of these resources into the application as it is generated.

The main procedure in the ADMS is a batch process that is executed when the "UPDATE" function is selected from the application's main pull-down menu. This process automates the task of updating membership data each month. It performs four distinct functions for the user [Ref. 1]. The first function is marking Alumni Association membership records that have expired. The "CURRENT" field in the database is used to indicate if the membership is expired or not. The second function performed by the process is the displaying of a screen that shows all the expired memberships to the user. This screen will allow the user to update memberships that have been renewed. This encompasses the third process. The user will update the "CURRENT" data field by entering a "Y" for all memberships that have been renewed. The fourth process will begin when the user has updated all the renewed memberships and exits the browse screen. Upon exiting the browse screen the process will reset the "RENEWED" field to "F", the "CURRENT" field to "T", and advance the Member's expiration date by one year. The final action is the storing of the updated data into the database.

The "PRINT" option on the pull-down menu allows the user to generate reports and form letters that are kept current by the user performing the "UPDATE" function. With these reports and form letters being dependent on the "UPDATE" function the user can generate reports that are accurate to the current day and form letters that are accurate for the current month. As stated earlier,

Table 7, Appendix B, has descriptions of the form letters (called reports in dBASE IV).

2. Testing

Testing is an important function that generally is inadequately done when implementation deadlines become compressed. This was not the case in the testing of the ADMS, the implementation of the application occurred on schedule. The primary method of testing was accomplished by providing controlled data and verifying the output of the application for correctness. This type of testing, black box testing, is not concerned with the internal processes of the application. The primary thrust of the testing is the verification of the correct outputs.

Testing was completed in a week, using test data, by the application's developer. After successful testing, the application was tested using approximately 1200 of the Alumni Association's records. This was the final phase of testing the application.

The application required minor adjustments to some of the calculations required in the "UPDATE" process due to minor implementation mistakes. The majority of the corrective work that the system required involved the formatting of the reports and form letters on the Association's printer. These problems were quickly resolved through additional research on the off-brand printer and its interaction with dBASE IV.

3. Installation

There are two primary methods of bringing a new system on-line. The preferred method is running the new system in parallel with the old system. This method provides the user with the least amount of risk but requires the resources to run two systems. The second method, which involves more risk, is the scraping of the old system and the immediate adoption of the new system. The risk in this

method arises from the possible failure of the new system with no back-up to rely on. Because the Alumni Association was not replacing an existing system, the second installation method was used to install the system.

Once the system was installed, the users could begin training on the system. Training was accomplished over a three month period. The system was demonstrated to the users by the developer. The first month accounting cycle was performed for the users so that they could learn the system's operating procedures and ask questions. The next two accounting cycles were performed by the users with the developer answering questions as they arose. File maintenance and other system features were covered during this training period. Appendix D, provides the User's Guide for the Alumni Association's Database Management System.

III. CONCLUSIONS

The Alumni Association Database Management System has been on-line since late May 1992. It has significantly reduced the time that the part-time staff spends on Alumni and membership record keeping. The monthly accounting cycle can be completed in less than one hour, which includes the printing of the membership dues notices for the current month. To date, the system has achieved the goals that it was intended to accomplish.

The user's requirements have been met; but as with most systems, new user requirements and needs develop as the current system heightens the user's awareness of additional benefits. The current system has been developed so that future enhancements should be easy to incorporate. Maintenance of applications is an ongoing process and modifications on the system should follow a structured maintenance program to ensure continued success with the system.

There are three system modifications that would improve the performance of the system. First, the purchase of a 386 or 486 IBM compatible computer would greatly enhance the response time of the system's "UPDATE" process. As the size of the database grows the turnaround time of the process will slow down with the current system. Upgrading the computing power will greatly enhance the systems performance with minimal cost. The second recommendation is the purchase of a tape back-up system for the application. Currently the system backs-up to the "A" drive. In the future, the size of the database will prohibit this because of the finite space available on a floppy disk. A tape back-up system would solve the growing space requirements of the database. This would require the outlay of additional funds but the cost is minimal compared to the benefits. The last recommendation is the upgrading of the dBASE IV version 1.1 software to the new dBASE IV

version 1.5 software. This should be available to the Alumni Association as an upgrade since they are registered owners of version 1.1. The advantage of version 1.5 is that it supports a mouse. This feature would be invaluable to the user during data entry. This would allow the user to go directly to the desired data field, thus saving many key strokes. Again, the cost of this enhancement is minimal.

APPENDIX A

REQUIREMENTS DOCUMENTATION

A. TABLE 1: OBJECT DIAGRAM

LAST-NAME
FIRST-NAME
MI
SERVICE
SSN
GRADUATION-DATE
RANK
DOB
ACTIVE
MEMBER-TYPE
ADDRESS-1
ADDRESS-2
ADDRESS-3
CITY
STATE
ZIP
COUNTRY
CURRICULUM
LAST-CONTACT
ARCHIVE
COMMENTS
JOIN-DATE
CERTIFICATE
EXPIRES
CURRENT
RENEWED

ALUMNI

B. TABLE 2: OBJECT DEFINITION

LAST-NAME: Last-name
FIRST-NAME: First-name
MI: Middle-initial
SERVICE: Branch-of-service
SSN: Social-security-number
GRADUATION-DATE: Graduation-date-from-NPGS
RANK: Service-rank
DOB: Date-of-birth
ACTIVE: Duty-status
MEMBER-TYPE: Membership-status
ADDRESS-1:   First-address-field
ADDRESS-2:   Second-address-field
ADDRESS-3:   Third-address-field
CITY: City
STATE: State
ZIP: Zip-code
COUNTRY: Country
CURRICULUM: Curriculum-code-at-NPGS
LAST-CONTACT: Last-contacted-date
ARCHIVE: Archive-record
COMMENTS: Comments-on-Alumni
JOIN-DATE: Date-joined-Alumni-Association
CERTIFICATE: Certificate-status
EXPIRES: Membership-expiration-date
CURRENT: Membership-status
RENEWED: Renewed-membership-status

C. TABLE 3:   ALUMNI DOMAIN DEFINITIONS

ACTIVE:
    Logical 1
    Default value "Y"
    Duty status of alumni

ADDRESS-1:
    Text 30
    First line of mailing address

ADDRESS-2:
    Text 30
    Second line of mailing address

ADDRESS-3:
    Text 30
    Third line of mailing address

ARCHIVE:
    Logical 1
    Default value "N"
    Deactivates a record from accounting cycle

CERTIFICATE:
    Logical 1
    Default value "N"
    Annotates the sending of membership certificates

CITY:
    Text 30
     City of mailing address

COMMENTS:
    Memo 10
    Records textual notes for current record

COUNTRY:
    Text 20
    Country of mailing address

CURRENT:
    Logical 1
    Annotates status of membership dues

CURRICULUM:
    Text 2
    Two digit NPGS curriculum code

TABLE 3 continued

DOB:
  Date Format (MM/DD/YY)
  Alumni's date-of-birth

EXPIRES:
  Date Format (MM/DD/YY)
  Date that membership expires

FIRST-NAME:
  Text 10
  Alumni's first name

GRADUATION-DATE:
  Date Format (MM/DD/YY)
  Graduation date from NPGS

JOIN-DATE:
  Date Format (MM/DD/YY)
  Date joined Alumni Association

LAST-CONTACT:
  Date Format (MM/DD/YY)
  Date Alumni was last contacted

LAST-NAME:
  Text 20
  Alumni's last name

MEMBER-TYPE:
  Logical 1
  Default value "N"
  Annotates member or non-member status

MI:
  Text 1
  Alumni's middle initial

RANK:
  Text 10
  Service rank and "(Ret)" if retired

RENEWED:
  Logical 1
  Flag for renewed memberships

TABLE 3 continued

SERVICE:
    Text 12
    Branch of military service

SSN:
    Text 9, mask XXX-XX-XXXX
    Alumni's social security number

STATE:
    Text 2
    State of mailing address

ZIP:
    Text 9, mask XXXXX-XXXX
    Zip code of mailing address

D. TABLE 4:   ALUMNI UPDATE MECHANISMS

I. Add new ALUMNI data:

   A. Inputs:
- Alumni personal data
- Membership data if joining Alumni Association

   B. Outputs:
- New ALUMNI object instance in database
- New screen for next record entry

   C. Processing notes:
- Last name, first name, and middle initial form unique record identifier
- Data should be entered into the database in the format that the user wishes it to appear on letters and reports
- Membership data fields are locked out when MEMBER-TYPE = "N"

   D. Volume:
- Approximately 1000 per FY

   E. Frequency:
- Quarterly

II. Edit data in ALUMNI:

   A. Inputs:
- Alumni name
- List of information to be changed
- Alumni object instance from database

   B. Outputs:
- Modified object instance to database

   C. Processing notes:
- Last name, first name, and middle initial must be valid to enter request
- Invalid last name, first name, and middle initial results in record not found and an opportunity to try again

   D. Frequency:
- As required

E. TABLE 5:   ALUMNI DISPLAY MECHANISMS

I.   Query on ALUMNI

    A.   Output description:
- Screen showing Alumni personal data and membership data

    B.   Source data:
- ALUMNI object
- Last name, first name, and middle initial entered by user

    C.   Processing notes:
- Used to obtain Alumni data for editing

    D.   Volume:
- 2-3 per week

    E.   Frequency:
- As required

II.   Membership Roster:

    A.   Output description:
- Report of Alumni Association Members

    B.   Source data:
- ALUMNI object

    C.   Processing notes:
- Report to be selected from menu option
- Names listed alphabetically
- Can be viewed on screen or printed

    D.   Volume:
- Semi-annually

III.   Welcome Letter:

    A.   Output description:
- Form letter to all newly joined members of the Alumni Association

    B.   Source data:
- ALUMNI object

    C.   Processing notes:
- Report to be selected from menu option
- Letters are ordered by zip code
- Can be viewed on screen or printed

    D.   Volume:
- Monthly

TABLE 5 continued

IV.   Renewal Letter:

    A.   Output description:
- Form letter to all members of the Alumni Association that have memberships that expire in the next month

    B.   Source data:
- ALUMNI object

    C.   Processing notes:
- Report to be selected from menu option
- Letters are ordered by zip code
- Can be viewed on screen or printed

    D.   Volume:
- Monthly

V. Overdue Letter:

    A.   Output description:
- Form letter to all members of the Alumni Association that have memberships that are expired

    B.   Source data:
- ALUMNI object

    C.   Processing notes:
- Report to be selected from menu option
- Letters are ordered by zip code
- Can be viewed on screen or printed

    D.   Volume:
- Monthly

VI.   Overdue listing:

    A.   Output description:
- Report of Alumni Association memberships that are overdue sixty (60) days or more

    B.   Source data:
- ALUMNI object

    C.   Processing notes:
- Report to be selected from menu option
- Names listed from most to least delinquent
- Can be viewed on screen or printed

    D.   Volume:
- Monthly

TABLE 5 continued

VII.   Browse ALUMNI:

  A. Output description:
     • Browse screen showing Alumni personal data and membership
      data
  B. Source data:
     • ALUMNI object
  C. Processing notes:
     • Browse to be selected from menu option
     • String search can be performed to locate a specific record
     • Records are ordered alphabetically
     • Records can not be edited
  D. Frequency:
     • As required

APPENDIX B

DATA DICTIONARY

A. TABLE 6: ALUMNIDB.DBF DATA ELEMENTS

ELEMENT	TYPE	WIDTH	DESCRIPTION
LNAME	CHAR	20	Last name
FNAME	CHAR	10	First name
MI	CHAR	1	Middle initial
SERVICE	CHAR	12	Branch of service
SSN	CHAR	9	Social security number
GRAD_DATE	DATE	8	Graduation date from NPGS
RANK	CHAR	10	Service rank
DOB	DATE	8	Alumni's date of birth
ACTIVE	LOGICAL	1	Duty status (Active/Inactive)
MEMBR_TYPE	LOGICAL	1	Indicates if Alumni is a member of the Alumni Association
ADDR1	CHAR	30	First line of mailing address
ADDR2	CHAR	30	Second line of mailing address
ADDR3	CHAR	30	Third line of mailing address
CITY	CHAR	30	City of mailing address
STATE	CHAR	2	State (two letter abbreviation) of mailing address
ZIP	CHAR	9	Zip code of mailing address
COUNTRY	CHAR	20	Country of mailing address

30

TABLE 6 continued

ELEMENT	TYPE	WIDTH	DESCRIPTION
CURRIC	CHAR	2	Two digit NPGS academic code
L_CONTACT	DATE	8	Date Alumni was last contacted
ARCHIVE	LOGICAL	1	Indicates if the record is to be used only for historical data
COMMENTS	MEMO	256	Text space provided for record notes
JOIN_DATE	DATE	8	Date Alumni joined the Alumni Association
CERT	LOGICAL	1	Indicates if membership certificate has been sent
EXPIRES	DATE	8	Membership expiration date
CURRENT	LOGICAL	1	Indicates if membership dues are current
RENEWED	LOGICAL	1	Flag for renewed memberships

B. TABLE 7:   ALUMNI ASSOCIATION DBMS REPORTS

REPORT FILE	VIEW FILE	DESCRIPTION
WELCOME.FRG	ALUMNIDB.DBF	WELCOME FORM LETTER: End of month letter sent to new Association members. Letters are grouped by zip code for bulk mailing rates.
RENEWAL.FRG	ALUMNIDB.DBF	RENEWAL FORM LETTER: Monthly letter sent to Association members with memberships that expire in the next month. Letters are grouped by zip code for bulk mailing rates.
OVERDUE.FRG	ALUMNIDB.DBF	OVERDUE FORM LETTER: Monthly letter sent to Association members with memberships that are expired. Letters are grouped by zip code for bulk mailing rates.
MEMLIST.FRG	ALUMNIDB.DBF	ALUMNI ASSOCIATION MEMBERSHIP ROSTER: List of Association members, branch of service, and join date. Names on the report are listed in alphabetic order.
PASTDUE.FRG	ALUMNIDB.DBF	PAST DUE LISTING: List of Association members who have membership dues delinquent more than sixty (60) days. Names on the list are ordered from most to least delinquent memberships.

APPENDIX C

ALUMNI ASSOCIATION DBMS APPLICATION DOCUMENTATION

Documentation for System: ALUMMGR.PRG

Application Author:: Mark Kohlheim

Display Application Sign-On Banner: Yes

Screen Image:

```
        0      10      20      30      40      50      60      70
>.....+....|....+....|....+....|....+....|....+....|....+....|....+....|....+.
00:
01:
02:
03:
04:
05:#==================================================
06:      "
07:      "             WELCOME TO THE
08:      "
09:      "         NAVAL POSTGRADUATE SCHOOL
10:      "
11:     ."           ALUMNI ASSOCIATION
12:      "
13:      "         DATABASE MANAGEMENT SYSTEM
14:      "
15:      "
16:#==================================================
17:
18:
19:
20:
21:
22:
23:
24:
>.....+....|....+....|....+....|....+....|....+....|....+....|....+....|....+.
```

Main Menu to Open after Sign-On: MAINMENU.BAR Application

Sets for Application:

```
------------------------------
    Bell           ON
    Carry          OFF
    Century        OFF
    Confirm        OFF
    Delimiters     OFF
    Display Size 25 lines
    Drive
    Escape         ON
    Path
    Safety         ON
```

Starting Colors for Application:

```
------------------------------
```

```
 Color Settings:
    Text           : W+/B
    Heading        : W+/B
    Highlight      : W+/BG
    Box            : BG+/B
    Messages       : W+/R
    Information    : W+/BG
    Fields         : W+/BG
```

Database/View: ALUMNIDB

Index Order: LASTNAME

```
===================================================================
```

Menu/Picklist definitions follow:

```
----------------------------------
```

Layout Report for Horizontal Bar Menu: MAINMENU

```
  ---------------------------------------------------------------------------
```

Screen Image:

```
0       10      20      30      40      50      60      70
>.....+....|....+....|....+....|....+....|....+....|....+....|....+....|....+.
00:#=================================================================
01:"Maintenance        Update         Print         Utilities            Exit
02:#=================================================================
03:
04:
05:
06:
07:
08:
```

```
09:
10:
11:
12:
13:
14:
15:
16:
17:
18:
19:
20:
21:
22:
23:
24: >.....+....|....+....|....+....|....+....|....+....|....+....|....+....|....+.
```

Setup for MAINMENU follows:

Description: ALUMNI APPLICATION MAIN MENU

Colors for Menu/Picklist:

 Color Settings:
   Text   : W+/G
   Heading   : W+/B
   Highlight   : GR+/GR
   Box   : N/W
   Messages   : W+/B
   Information   : GR+/BG
   ·Fields   : GR+/BG

Bar actions for Menu MAINMENU follow:

Bar: 1
 Prompt: Maintenance
 Action: Open a Popup Menu Named: MAINTMNU

Bar: 2
 Prompt: Update
 Action: Open a Popup Menu Named: UPDATMNU

Bar: 3
 Prompt: Print

Action: Open a Popup Menu Named: PRINTMNU

Bar: 4
 Prompt: Utilities
 Action: Open a Popup Menu Named: UTILSMNU

Bar: 5
 Prompt: Exit
 Action: Open a Popup Menu Named: EXITMENU

Layout Report for Popup Menu: MAINTMNU

Screen Image:

```
        0        10                20      30      40      50      60      70
>.....+....|....+....|....+....|....+....|....+....|....+....|....+....|....+.
00:
01:
02:    #===============================#
03:    " Add an Alumni                 "
04:    " Browse all Alumni             "
05:    " Edit/Delete an Alumni's Data  "
06:    #===============================#
07:
08:
09:
10:
11:
12:
13:
14:
15:
16:
17:
18:
19:
20:
21:
22:
23:
```

24: >.....+....|....+....|....+....|....+....|....+....|....+....|....+....|....+.

Setup for MAINTMNU follows:

-----------------------:--

Description: MAIN MAINTENANCE POP-UP MENU
Colors for Menu/Picklist:

Color Settings:
    Text   : W+/B
    Heading   : W+/B
    Highlight   : GR+/GR
    Box   : N/W
    Messages   : W+/B
    Information   : W+/B
    Fields   : W+/B
Bar actions for Menu MAINTMNU follow:

Bar: 1
    Prompt: Add an Alumni
    Action: APPEND
    Format File: mainform.fmt
    Position Record Pointer by: GOTO bottom

 Help text defined for this item:

F10 : File Operations Menu Cntrl-End : Save Changes
    Esc : Exit (no changes will be saved)
    NOTE
    Tabbing through the end of the record to the next record will save changes.

Bar: 2
 Prompt: Browse all Alumni
 Action: Browse File
 Command Options:
   NOAPPEND NODELETE NOEDIT
   FORMAT
 Format File: mainform.fmt
 Position Record Pointer by: GOTO top

Position Record Pointer by: GOTO top
Help text defined for this item:

    F10 : File Operations Menu (browse functions only, no editing)
    Esc : Exit
    NOTE
    This is a browse only area. No editing can be performed in this area.

Bar: 3
Prompt: Edit/Delete an Alumni's Data
Action: EDIT
Format File: mainform.fmt
Position Record Pointer by: GOTO top
Help text defined for this item:

    F10 : File Operation Menu
    Cntrl-End : Save Changes
    Esc : Exit (no changes will be saved)
    NOTE
    Tabbing through the end of the record to the next record also saves changes.

Layout Report for Popup Menu: UPDATMNU

Screen Image:

```
        0        10              20      30      40      50      60      70
>.....+....|....+....|....+....|....+....|....+....|....+....|....+....|....+.
00:
01:
02:            #=============================#
03:            " Update Membership Renewals "
04:            #=============================#
05:
06:
07:
08:
09:
10:
```

38

```
11:
12:
13:
14:
15:
16:
17:
18:
19:
20:
21:
22:
23:
24: >.....+....|....+....|....+....|....+....|....+....|....+....|....+....|....+.
```

Setup for UPDATMNU follows:

Description: Use Post Dues Payments
Colors for Menu/Picklist:

Color Settings:
    Text   : W+/B
    Heading   : W/B
    Highlight   : GR+/GR
    Box   : N/W
    Messages   : W+/B
    Information   : N/W
    Fields   : N/W
Bar actions for Menu UPDATMNU follow:

Bar: 1
Prompt: Update Membership Renewals
Action: Call Batch Named: RENEW

Layout Report for Popup Menu: PRINTMNU

Screen Image:

   0   10   20   30   40   50   60   70

```
>.....+....|....+....|....+....|....+....|....+....|....+....|....+....|....+.
00:
01:
02:          #=======================#
03:          " Membership Roster"
04:          " All Mailing Labels"
05:          " Welcome Letter"
06:          " Welcome Labels"
07:          " Renewal Letters"
08:          " Renewal Labels"
09:          " Overdue Letters"
10:          " Overdue Labels"
11:          " Print Overdue listing  "
12:          · "_____"
13:          "Complete Updates in"
14:          "the UPDATE MENU before  "
15:          "printing Monthly Letters"
16:          #=======================#
17:
18:
19:
20:
21:
22:
23:
24:
>.....+....|....+....|....+....|....+....|....+....|....+....|....+....|....+.
```

Setup for PRINTMNU follows:

Description: Use this Menu to Print Form Letters and Mailing Labels
Colors for Menu/Picklist:

Color Settings:
    Text   : W+/B
    Heading   : W/B
    Highlight   : GR+/GR
    Box   : N/W
    Messages   : W+/B
    Information   : N/W
    Fields   : N/W

Bar actions for Menu PRINTMNU follow:

Bar: 1
Prompt: Membership Roster
Action: Run Report Form MEMLIST.frm
Command Options:
    PLAIN
Print Mode: Ask User at RuntimeFilter:
MEMBR_TYPE = .Y. .AND. ARCHIVE <> .Y.
Before dBASE Code for this item:

    _PLENGTH=60
    _PLOFFSET=5

Bar: 2
Prompt: All Mailing Labels
Action: Run Label Form MEMLABEL.lbl
Print Mode: Ask User at Runtime
Filter: MEMBR_TYPE = .Y. .AND.
ARCHIVE <> .Y.
Set Order To ZIP
Before dBASE Code for this item:

    _ploffset=0
    _peject="after"

Bar: 3
 Prompt: Welcome Letter
 Action: Run Report Form WELCOME.frm
 Command Options:
    FOR MONTH(JOIN_DATE) = MONTH(DATE()) .AND.
YEAR(JOIN_DATE)=YEAR(DATE())
Print Mode: Ask User at RuntimeFilter:
MEMBR_TYPE = .Y. .AND. ARCHIVE <> .Y.
 Set Order To ZIP
 Before dBASE Code for this item:

    _PLOFFSET=5
    _peject="after"

41

Bar: 4
 Prompt: Welcome Labels
 Action: Run Label Form MEMLABEL.lbl
 Command Options:
   FOR MONTH(JOIN_DATE)=MONTH(DATE()) .AND.
YEAR(JOIN_DATE)=YEAR(DATE())
 Print Mode: Ask User at Runtime
 Set Order To ZIP
 Before dBASE Code for this item:

   _ploffset=0
   _peject="after"

Bar: 5
 Prompt: Renewal Letters
 Action: Run Report Form RENEWAL.frm
 Command Options:

FOR MONTH(Expires)=MOD(MONTH(DATE()),12)+1 .and. archive <>.y.

Print Mode: Ask User at RuntimeFilter:
YEAR(Expires)=YEAR(DATE())+INT(MONTH(EXPIRES)/12)
.AND.MEMBR_TYPE=.Y.
 Set Order To ZIP
 Before dBASE Code for this item:

   _PLOFFSET=5
   _peject="after"

Bar: 6
 Prompt: Renewal Labels
 Action: Run Label Form MEMLABEL.lbl
 Command Options:
   FOR MONTH(Expires)=MOD(MONTH(DATE()),12)+1 .and. archive <> .y.
 Print Mode: Ask User at Runtime
Filter:
YEAR(Expires)=YEAR(DATE())+INT(MONTH(Expires)/12)

.AND.MEMBR_TYPE=.Y.
Set Order To ZIP
Before dBASE Code for this item:

    _ploffset=0
    _peject="after"

Bar: 7
Prompt: Overdue Letters
Action: Run Report Form OVERDUE.frm
Command Options:
   FOR DATE()-Expires>30 .AND. .NOT. Current
Print Mode:
Ask User at RuntimeFilter: MEMBR_TYPE=.Y. .AND. ARCHIVE<>.Y.
Set Order To ZIP
Before dBASE Code for this item:

    _PLOFFSET=5
    _peject="after"

Bar: 8
Prompt: Overdue Labels
Action: Run Label Form MEMLABEL.lbl
Command Options:
   FOR DATE()-Expires>30 .AND. .NOT. Current
Print Mode: Ask User at Runtime
Filter: MEMBR_TYPE=.Y. .AND. ARCHIVE<>.Y.
Set Order To ZIP
Before dBASE Code for this item:

    _ploffset=0
    _peject="after"
After dBASE Code for this item:

----

    eject

Bar: 9
 Prompt: Print Overdue listing
 Action: Run Report Form PASTDUE.frm
 Command Options:
   FOR DATE()-EXPIRES>60 .AND. .NOT. CURRENT
   NOEJECT
Print Mode: Ask User at RuntimeFilter: MEMBR_TYPE=.Y. .AND.
ARCHIVE<>.Y.
 Set Order To EXPIRES
 Before dBASE Code for this item:

   _PLENGTH=60
   _PEJECT="NONE"
   _ploffset=5

Bar: 10
 Prompt: _____
 Action: Text only defined for this option - NO ACTION

Bar: 11
 Prompt: Complete Updates in
 Action: Text only defined for this option - NO ACTION

Bar: 12
 Prompt: the UPDATE MENU before
 Action: Text only defined for this option - NO ACTION

Bar: 13
 Prompt: printing Monthly Letters
 Action: Text only defined for this option - NO ACTION

Layout Report for Popup Menu: UTILSMNU

44

Screen Image:

```
        0         10                  20        30        40        50        60        70
>.....+....l....+....l....+....l....+....l....+....l....+....l....+....l....+.
00:
01:
02:               #========================#
03:               " Change System Date        "
04:               " Backup Database to Drive A  "
05:               " Copy Application to Drive A "
06:               " Repair Corrupted Indexes      "
07:               #========================#
08:
09:
10:
11:
12:
13:
14:
15:
16:
17:
18:
19:
20:
21:
22:
23:
24:
>.....+....l....+....l....+....l....+....l....+....l....+....l....+....l....+.
```

Setup for UTILSMNU follows:

Description: Use this menu to Backup, change date, and repair indexes
Colors for Menu/Picklist:

Color Settings:
   Text   : W+/B
   Heading   : W/B
   Highlight   : GR+/GR
   Box   : N/W
   Messages   : W+/B
   Information   : N/W
   Fields   : N/W

45

Bar actions for Menu UTILSMNU follow:

Bar: 1
 Prompt: Change System Date
 Action: Run Dos Program - DATE

Bar: 2
 Prompt: Backup Database to Drive A
 Action: Run Dos Program - COPY ALUMNIDB.* A:

Bar: 3
 Prompt: Copy Application to Drive A
 Action: Run Dos Program - COPY *.* A:

Bar: 4
 Prompt: Repair Corrupted Indexes
 Action: Reindex Current DBF:

Layout Report for Popup Menu: EXITMENU

Screen Image:

```
      0        10             20      30      40      50      60      70
>.....+....|....+....|....+....|....+....|....+....|....+....|....+....|....+.
00:
01:
02:                         #====================#
03:                         " Return to dBASE IV "
04:                         " Exit to DOS        "
05:                       . #====================#
06:
07:
08:
09:
10:
```

```
11:
12:
13:
14:
15:
16:
17:
18:
19:
20:
21:
22:
23:
24: >.....+....|....+....|....+....|....+....|....+....|....+....|....+....|....+.
```

Setup for EXITMENU follows:

Description: Use this Menu to Exit this Application
Colors for Menu/Picklist:

Color Settings:
    Text   : W+/B
    Heading   : W/B
    Highlight   : GR+/GR
    Box   : N/W
    Messages   : W+/B
    Information   : N/W
    Fields   : N/W
Bar actions for Menu EXITMENU follow:

Bar: 1
 Prompt: Return to dBASE IV
 Action: Return to calling program

Bar: 2
 Prompt: Exit to DOS
 Action: Quit to DOS:

Multiple Action Summary for Batch Object: RENEW

Screen Image:

```
          0         10                20    30    40    50    60    70
>.....+....|....+....|....+....|....+....|....+....|....+....|....+....|....+.
00:
01:
02:
03:
04:
05:         #========================#
06:         " Flag Expired Memberships        "
07:         " Browse Expired Memberships     "
08:         " Update Renewed Memberships  "
09:         " Reset RENEWED Field value to .F. "
10:         #========================#
11:
12:
13:
14:
15:
16:
17:
18:
19:
20:
21:
22:
23:
24:
>.....+....|....+....|....+....|....+....|....+....|....+....|....+....|....+.
```

Setup for RENEW follows:

Description: Automated monthly renewals
Colors for Menu/Picklist:

Color Settings:
    Text   : W+/B
    Heading   : W/B
    Highlight   : BG+/R
    Box   : BG+/R
    Messages   : W+/R

```
    Information           : BG+/R
    Fields                : BG+/R
Batch actions for Menu RENEW follow:
```

Batch Action: 1
 Prompt: Flag Expired Memberships
 Action: Replace Records in
 Command Options:
   Fields CURRENT WITH .F.
FOR MONTH(Expires)<=MONTH(DATE()) .AND. YEAR(Expires)<=
YEAR(DATE())

   Before dBASE Code for this item:

   MEMBR_TYPE=.Y. .AND. ARCHIVE<>.Y.

Batch Action: 2·
 Prompt: Browse Expired Memberships
 Action: Browse File
 Command Options:

FIELDS RENEWED,LNAME,FNAME,RANK,JOIN_DATE,EXPIRES,
CURRENT WIDTH 15 FREEZE RENEWED NOAPPEND NODELETE
 Filter: .NOT. CURRENT .AND. MEMBR_TYPE=.Y. .AND. ARCHIVE<>.Y.
   Before dBASE Code for this item:

   MEMBR_TYPE=.Y. .AND. ARCHIVE<>.Y.

Batch Action: 3
 Prompt: Update Renewed Memberships
 Action: Replace Records in
 Command Options:
Fields EXPIRES WITH EXPIRES+365, CURRENT WITH .Y. FOR
MEMBR_TYPE=.Y. .AND. ARCHIVE<>.Y. .AND. RENEWED
 Before dBASE Code for this item:

MEMBR_TYPE=.Y. .AND. ARCHIVE<>.Y.

Batch Action: 4
 Prompt: Reset RENEWED Field value to .F.

Action: Replace Records in
Command Options:
  Fields RENEWED WITH .N.
  SCOPE ALL FOR MEMBR_TYPE=.Y. .AND. ARCHIVE<>.Y.
  Before dBASE Code for this item:

  MEMBR_TYPE=.Y. .AND. ARCHIVE<>.Y.

End of Application Documentation

APPENDIX D

ALUMNI ASSOCIATION DBMS USER'S GUIDE

A. INTRODUCTION

This guide is intended to be used by individuals who have a basic understanding of the use and operations of Ashton-Tate's dBase IV program. The Alumni Association database management system (DBMS) application and its related files should be located in its own sub-directory. The application was originally set-up on the Alumni Association's computer system in its own sub-directory within the dBase IV directory. The DOS path and directory names are as follows:

C:\dBase\Alumni

B. STARTING THE APPLICATION

A batch program has been written and installed on the root directory of the "C" drive that will start dBase IV and the ALUMMGR application from the DOS C:/ prompt. The ALUMMGR application is started by executing the following steps:

Step 1:   Type "ALUMMGR" at the DOS C:/ prompt

Step 2:   Press "ENTER" at the dBASE copyright screen

C.   DATA ENTRY AND NAVIGATION CONVENTIONS

Data can be entered or edited by typing the information desired into the appropriate data field. If all spaces within a field are used, then the cursor will automatically advance to the next field. The arrow keys (← →) will advance the cursor one space forward or backward one space at a time and to the next field if the cursor is located at the end of a field. To move the cursor forward or backward

51

a field at a time can be accomplished by using the TAB key to move forward and SHIFT-TAB to move backward. Tabbing through the last data field on a data entry screen will automatically save any changes and will advance to the next logical screen. Saving changes and moving forward and backward a screen at a time can also be accomplished by using the PAGE-UP and PAGE-DOWN keys.

Menus can be navigated by using the arrow keys ($\leftarrow\downarrow\uparrow\rightarrow$). Moving horizontally across the main application menu will automatically display a pop-up menu associated with the highlighted option. Pressing ENTER will activate any option selected within any of the pop-up menus.

D. OPERATION

Once the application is started, the user will be greeted by a "Welcome Screen." The screen should be identical to the one in Figure D.1. If it does not appear or if there is any other discrepancy the user should review the start-up procedures and attempt to restart the application.
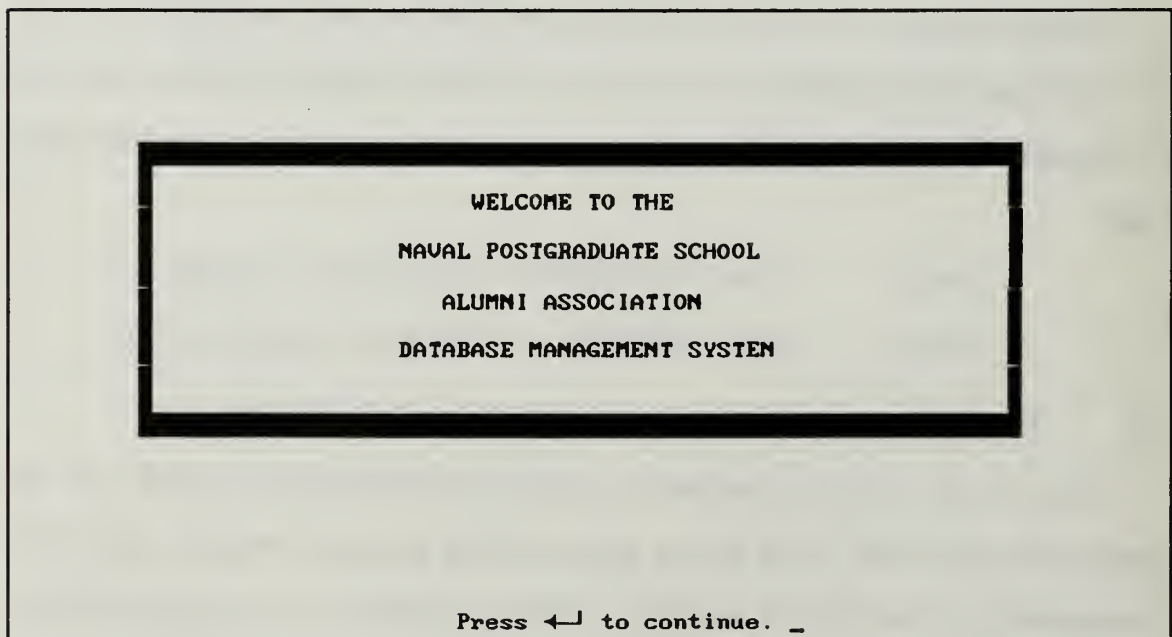


```
                    WELCOME TO THE

               NAVAL POSTGRADUATE SCHOOL

                  ALUMNI ASSOCIATION

              DATABASE MANAGEMENT SYSTEM




              Press ←┘ to continue. _
```

Figure D.1   Introduction Screen

E. MAIN MENU

The main menu of the application will appear after the sign-on screen is deactivated by pressing ENTER. The pop-up menu for "Maintenance" will be automatically activated as shown in Figure D.2. Pressing the left or right arrow keys (← →) will activate the pop-up menu for the option selected. The up and down arrow keys (↑↓) can be used to select an option within the selected pop-up menu.
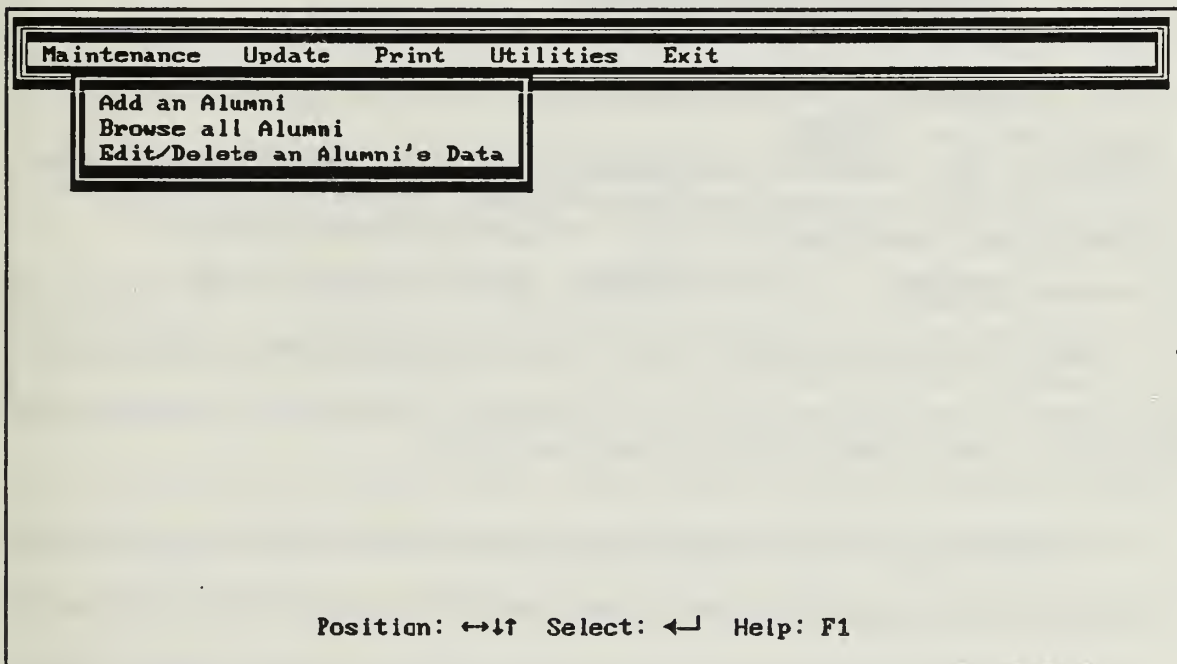
```
╔══════════════════════════════════════════════════════════════════════╗
║ Maintenance   Update    Print   Utilities    Exit                      ║
╚══════════════════════════════════════════════════════════════════════╝
  ┌──────────────────────────────────┐
  │ Add an Alumni                    │
  │ Browse all Alumni                │
  │ Edit/Delete an Alumni's Data     │
  └──────────────────────────────────┘




                   Position: ↔↓↑  Select: ↵  Help: F1
```

Figure D.2   Main Menu / Maintenance Pop-Up Menu

F. MAINTENANCE MENU

The three file maintenance options available in the maintenance menu are shown in Figure D.2.

1.   Add an Alumni

This option presents a blank data entry screen for the user to enter new Alumni to the database. This screen is shown in Figure D.3. Some data fields have default values present to assist in the entering of generic alumni data. These

values can be modified by entering the desired data and over writing the default value. Data field masks are also provided for the ease of data entry as well as for easier viewing on the screen.

Figure D.3   Add Alumni / Edit Alumni Data Screen

Data entry can be aborted without saving the new record by pressing the ESC key. The new record can be saved by tabbing through the end of the data entry screen which will also give the user another blank data entry screen. CTRL-END can also be pressed to save the new record and will return the user to the maintenance pop-up menu.

a.   Data Fields and Data Entry

This section will step through all of the data fields on the "Add Alumni" screen and describe the function of each field as well as any specific formatting requirements for the data.

"Rank" provides ten (10) spaces to enter any service rank. Normally the alumni's rank will be entered using the capitalized initials for the desired rank.

"First Name" provides ten (10) spaces to enter the first name of the alumni. "MI" is for the middle initial and is one space. "Last Name" provides twenty-five (25) spaces for the last name of the alumni. The first letter of the first and last name should be capitalized and the remaining letters should be in lower case. The letter entered for the middle initial should also be capitalized.

"Service" provides twelve (12) characters for the entering of the service of the alumni. If the individual is retired this should be annotated in this field by entering "(Ret)" after the branch of service entered. "Active" is a Boolean field that is used to distinguish active from retired service members. A "Y" should be entered for active service and a "N" should be entered for retired service members.

"Address" provides three lines which contain thirty (30) spaces each. These three lines provide enough space to enter an Alumni's address and additional codes. Normally the address should start at the top line and only the lines necessary for the address should be used. "State" provides two (2) spaces for the letter code for the address state. These letters should be entered as capital letters. "Zip" is a numeric field that will accept either a five or nine digit zip code. "Country" provides twenty (20) spaces for the entry of the country if it is required for the address.

"SSN" field is for the nine digit social security number of the alumni. "DOB" is for the eight digit birth date using a date format of "MM/DD/YY". "Graduation Date" is for the eight digit date of the alumni's graduation. "Curriculum" provides for the two (2) digit NPGS academic code that the Alumni attended. The "Comments (F9)" field provides for the entry of text (up to 256 characters) information if required. The field will display "memo" if there is no textual information entered into this field and "MEMO" if there is textual information entered in this field. To activate the field to enter or read the text it

contains, press "F9." "Last Contact" is a field provided to enter the last date that the Alumni was contacted. The field provides eight (8) spaces to enter the date in the format of "MM/DD/YY." The "Archive" field is a Boolean field that allows the user to indicate that an Alumni's record is inactive. The default setting for this field is "N." This will allow the record to be used for historical data but not for current billing and mailing operations. If the record is to be made inactive (archived), enter "Y" into the field.

"Alumni Association Member (Y/N)" is a Boolean field that allows the user to activate an alumni as an active Alumni Association member. The default setting is "N." When the field is set to "N" none of the remaining screen fields can be accessed by the user. When the field is set to "Y" the remaining screen fields can be accessed and have data entered into them. "Dues Current" is a Boolean field that should be initially set to "Y" when an alumni joins the association. After the initial setting, the field will be updated by the application and no further user manipulation is required.

"Join Date" provides eight (8) spaces that allows the user to enter the joining date for the Alumni using the format "MM/DD/YY." This field will always remain the same as long as the Alumni remains a member of the Association. "Expires" provides eight (8) spaces that allows the user to enter the first expiration date of the membership (MM/DD/YY). This date should be one year from the join date initially. The expiration date will be updated by the application whenever the member pays his/her annual dues and it is entered into the database. The last field, "Certificate Sent," is a Boolean that is initially set to "N" by default. This field allows the user to keep track of membership certificates being sent to new members. An "Y" indicates that a certificate has been sent to the member.

2.   Browse all Alumni

This option will display a normal dBase IV browse screen which will allow viewing of alumni records. The records are organized in alphabetical order and can be scrolled through by using the arrow keys (←↑↓→) and the PAGE-UP / PAGE-DOWN keys. The browse alumni option is for viewing only, therefore, editing of records cannot be performed using this option.

3.   Edit / Delete an Alumni's Data

Selection of this option will allow the user to edit and delete individual database records. When this selection is chosen, the user will be presented with an identical screen as the one used in the Add an Alumni option (Figure D.3). The screen will display the first alphabetically ordered record stored in the database. The user can move through the database one record at a time using the PAGE-UP / PAGE-DOWN keys. Selection of a specific record in the database can be accomplished by conducting an index search. To conduct an index search the user will press F10 key to obtain the normal dBase IV file maintenance menu bar. The "GOTO" option will be selected next and then the "Index Key Search" option will be selected. This menu is displayed in Figure D.4.

```
Records      Organize      Go To     Exit

                         Top record
                         Last record
                         Record  number         {1127}
                         Skip                   {10}

                         Index key search       {}
                         Forward search         {}
                         Backward search        {}
                         Match capitalization   YES
```

Figure D.4   Index Key Search

At this point the user will enter the last name of the alumni record desired and then press "ENTER." Correct spelling and capitalization of the name entered is important for selection of the correct record. If the record exists in the database the screen will display the requested record and if there is no match to the name entered the user will be presented with a "Record Not Found" message.

Editing a record can be accomplished using the same conventions used in adding a record to the database. Deleting a record can be done by using dBase IV's normal method of marking records for deletion and then deleting the marked records. The deletion of a record can be accomplished using the same menu shown in Figure D.4 and selecting the "Records" and "Organize" selections to mark and delete records. The "Exit" selection or the ESC key will hide the file maintenance menu and return the user back to the edit screen.

a. Data Fields and Data Entry

The same field descriptions and data entry conventions should be used as outlined in Section F.1.a.

G. UPDATE MENU

The "Update Menu" executes a batch process which performs four distinct actions for the user. The first action of the batch process is to search the database and flag all expired memberships. The second process is to present the expired memberships in a modified browse screen which is presented in Figure D.5. Expired memberships are presented in alphabetical order with the cursor locked in the renewed field. The renewed field is the only item that can be edited in this screen.

RENEWED	LNAME	FNAME	RANK	JOIN_DATE	EXPIRES	CURRENT
E	Duyn	Wade	CDR	01/10/92	01/09/85	F

Figure D.5   Expired Memberships Browse Screen

The user will renew memberships in the database by typing a "Y" into the renewed column for the Alumni members who have paid their annual dues. When all of the update entries have been made for the dues received, the user will continue the execution of the batch process by pressing the ESC key. No further action is required by the user.

The batch process will complete its final two steps at this time. All renewed memberships will be marked as current and the expiration data of renewed memberships will be advanced one year. Upon completion of the "UPDATE" batch process, the application will return to the main application menu.

H. PRINT MENU

The "Print Menu" presents the user with all of the reports and form letter preparations that the Alumni Association uses on a recurring basis. This menu selection is presented in Figure D.6.

```
 Maintenance    Update    Print    Utilities    Exit
┌─────────────────────────────┐
│        Membership Roster     │
│        All Mailing Labels    │
│        Welcome Letter        │
│        Welcome Labels        │
│        Renewal Letters       │
│        Renewal Labels        │
│        Overdue Letters       │
│        Overdue Labels        │
│        Print Overdue listing │
│                              │
│ Complete Updates in          │
│ the UPDATE MENU before       │
│ printing Monthly Letters     │
└─────────────────────────────┘



              Position: ↔↓↑  Select: ←┘  Help: F1
```
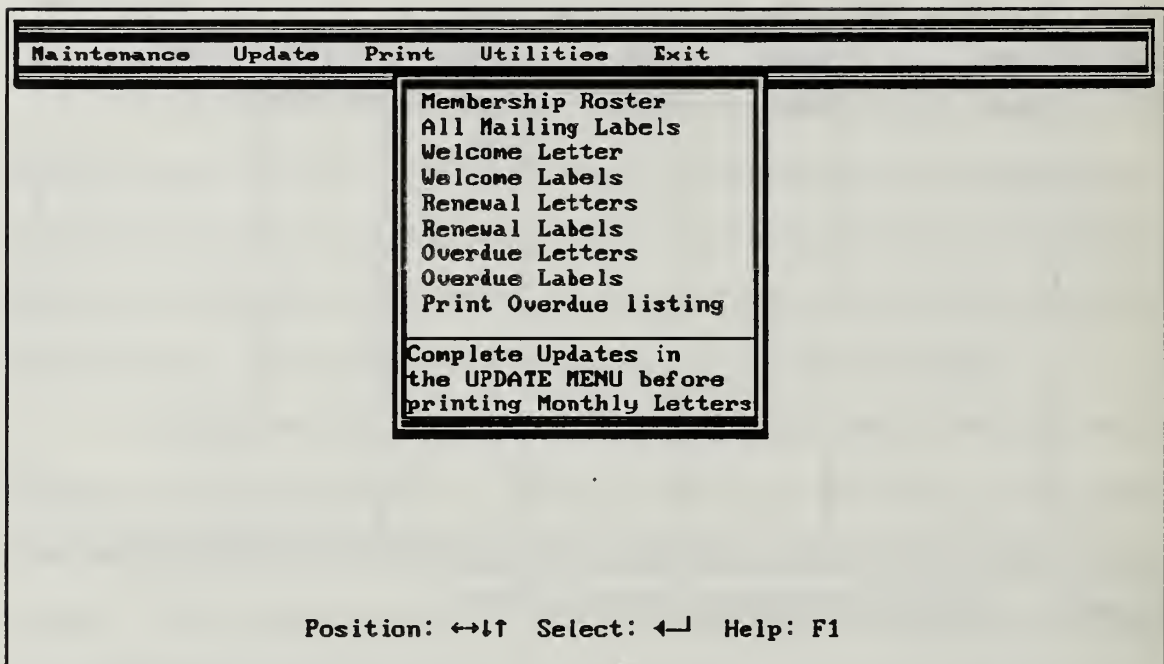
Figure D.6   Print Menu

It is important that the user perform the update function discussed in Section F
(even when no updates are to be entered) so that letters and reports prepared are
current through the date of preparation. A special note is presented at the bottom
of the menu selection to help remind the user to perform the update function.

For any of the print options selected, the user will be presented with a pop-up
menu (shown in Figure D.7) that will allow the user to select the output port for
the print function. The two normal output selections will be "Console" and
"LPT1". Selecting "Console" will direct the output of the print function selected
to the computer screen for the user to view. Selecting "LPT1" will send the print
function to printer port one. While the print job is being printed the output will
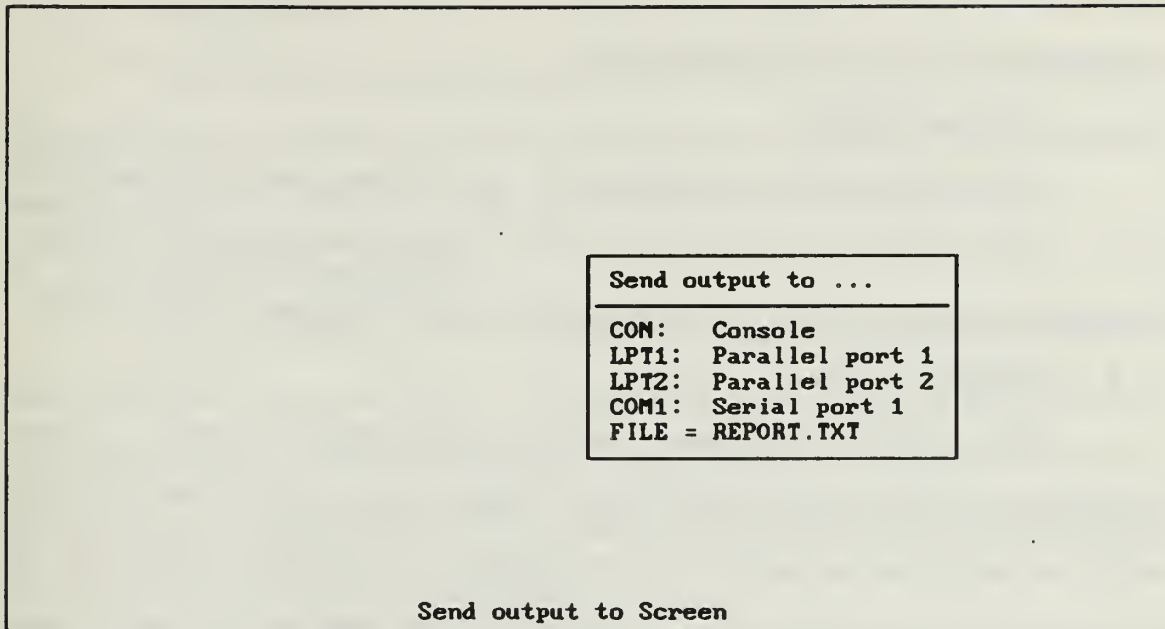also be displayed on the computer screen for the user to view.

```
┌──────────────────────────────────────────────────────────────┐
│                                                              │
│                                                              │
│                                                              │
│                                                              │
│                                                              │
│                  ┌─────────────────────────────┐             │
│                  │ Send output to ...          │             │
│                  ├─────────────────────────────┤             │
│                  │ CON:    Console             │             │
│                  │ LPT1:   Parallel port 1     │             │
│                  │ LPT2:   Parallel port 2     │             │
│                  │ COM1:   Serial port 1       │             │
│                  │ FILE =  REPORT.TXT          │             │
│                  └─────────────────────────────┘             │
│                                                              │
│                                                              │
│                  Send output to Screen                       │
└──────────────────────────────────────────────────────────────┘
```

Figure D.7   Print Output Selection Menu

1.   Print Membership Roster

This selection will print a current listing of all the members of the Alumni Association in alphabetical order.

2.   All Mailing Labels

This selection will print mailing labels for all of the current Alumni Association members. The mailing labels are sorted by zip code to meet the requirements for postal bulk mailing rates.

3.   Welcome Letter

This selection prints the "Welcome Letter" for all new members that have joined within the last month. The user has to exercise care that this function is completed at the end of the month. This selection will only print letters to new members that have joined within the last calendar month. Printing "Welcome Letters" should be scheduled at the end of the month. The entering of new members after the current month's. printing should be postponed until the

beginning of the next month. These letters are sorted by zip code to meet the requirements for postal bulk mailing rates.

4. Welcome Labels

This selection prints the labels for the welcome letters printed each month. These labels are sorted by zip code to meet the requirements for postal bulk mailing rates and will be in the same order as the "Welcome Letters" printed.

5. Renewal Letters

This selection will print renewal notices to all members whose memberships expire during the next month. These letters can be printed any time during the month, but normally should be mailed by the middle of the month to provide an adequate payment cycle. These letters are sorted by zip code to meet the requirements for postal bulk mailing rates.

6. Renewal Labels

This selection prints the labels for the renewal letters printed each month. These labels are sorted by zip code to meet the requirements for postal bulk mailing rates and will be in the same order as the renewal notices printed.

7. Overdue Letters

This selection prints overdue notices to members with membership expiration dates that are more than thirty days past due. These letters can be printed any time during the month. A regular schedule should be established for the preparation and mailing of these notices. The middle of the month would be an ideal time. These letters are sorted by zip code to meet the requirements for postal bulk mailing rates.

8. Overdue Labels

This selection prints the labels for the overdue letters printed each month. These labels are sorted by zip code to meet the requirements for postal bulk mailing rates and will be in the same order as the overdue notices printed.

9. Print Overdue Listing

This option prints a list of members with expired memberships which are more than sixty days past due. The list's output is ordered so that the oldest delinquent account is listed first and the latest delinquent account is listed last. This list can be useful to the user as a management tool for deciding which past due memberships should be deactivated.

I. UTILITIES MENU

The "Utilities Menu" provides the user with some file maintenance capabilities and a DOS interface that will allow modification of the operating system's date. The menu is shown in Figure D.8.
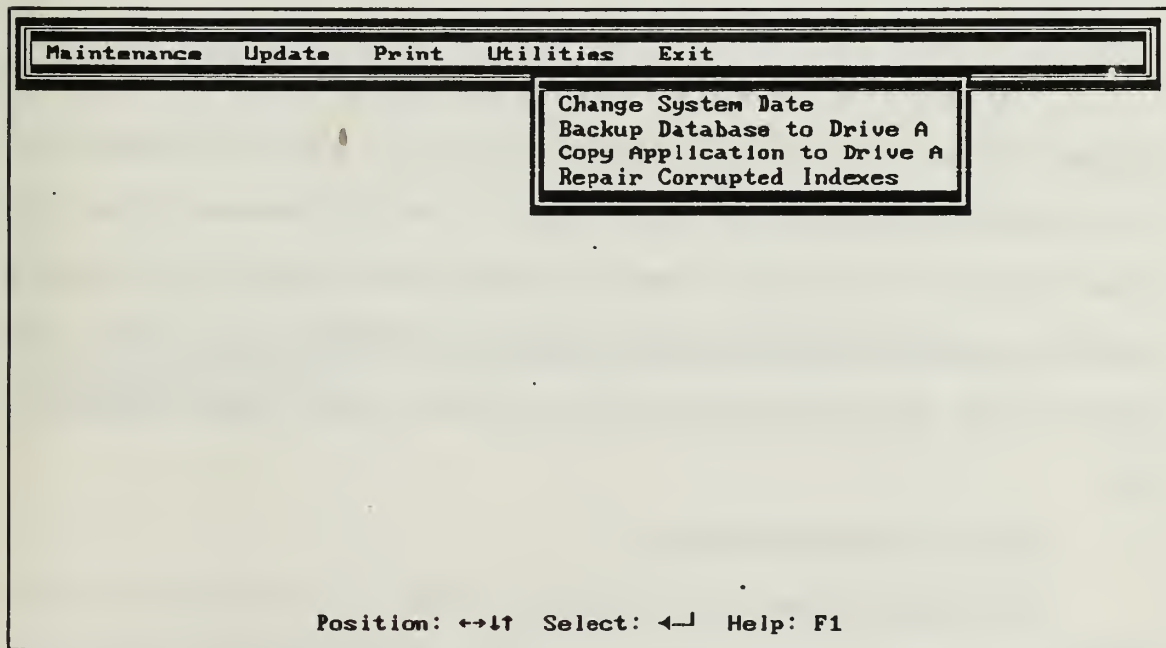

Figure D.8   Utilities Menu

1.   **Change system Date**

This option will allow the user to modify the system date while remaining in the application. This function will be useful to the user when last month's notices and reports have not been completed and a new month has begun. By back dating the system date, the user will be able to complete all of last month's transactions that are based on the current calendar month (these were discussed under the "Print Menu" section).

<u>CAUTION</u>:   *Change the system date back to the current date before attempting any other program functions.*

2.   **Backup Database to Drive A**

This option will backup the data file used for this application to the "A" drive of the computer system. This option should be executed before quitting the application whenever changes have been made to the data file. It is also recommended that the user make two or more copies of the data file to provide insurance against disk failure and other disk mishaps.

3.   **Copy Application to Drive A**

This option will backup the data file and application files used for this application to the "A" drive of the computer system. It is also recommended that the user make two or more copies of the data file to provide insurance against disk failure and other disk mishaps. One of the application backup files should be stored in a secured area away from any hazards that could damage the disk or the data on the disk. This disk should be used as duplicate for the original application disk.

4.   **Repair Corrupted Indexes**

This option repairs any corrupted indexes in the data file of the application. The application's indexes will get corrupted occasionally and is not an

indication of errors or misuse, it is only a by-product of using the data files extensively. This function should be executed any time the application appears to be running slowly (accessing data) or when other indexing functions do not operate properly. The only action required of the user is to activate this option and the application will perform the necessary actions.

J: EXIT MENU

The "Exit Menu" provides two paths of escaping from the application. The first path is exiting the application to the control panel in dBase IV which the application was launched. The second path is exiting out of dBase IV to the DOS "C:" prompt. The "Exit Menu" is shown in Figure D.9.
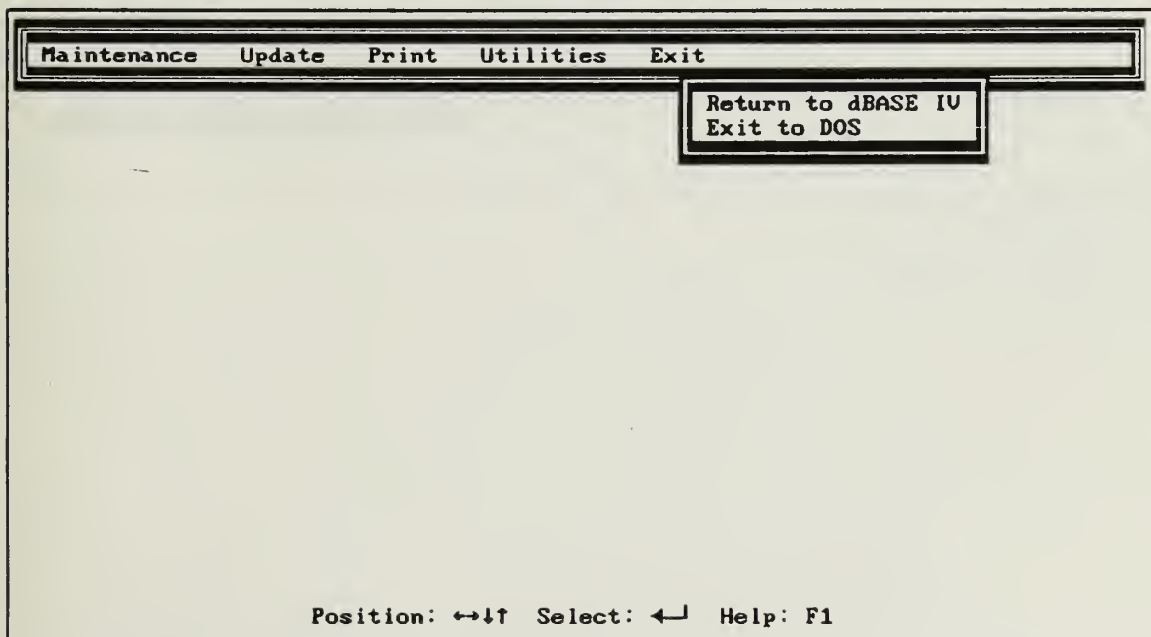

Figure D.9   Exit Menu

LIST OF REFERENCES

1. Simpson, A., *Understanding dBASE IV 1.1*, 2nd ed., Sybex, Inc., 1990.

BIBLIOGRAPHY

Kroenke, D.M. and Dolan, K.A., *Database Processing; Fundamentals, Design, Implementation*, 3rd ed., Macmillan Publishing Co., 1988.

Page-Jones, M., *Practical Guide to Structured Systems Design*, 2nd ed., Prentice-Hall, Inc., 1988.

Senn, J.A., *Analysis and Design of Information Systems*, 2nd ed., McGraw Hill, Inc., 1989.

Simpson, A., *dBASE IV Programmer's Instant Reference*, Sybex, Inc., 1989.

Simpson, A., *Understanding dBASE IV 1.1*, 2nd ed., Sybex, Inc., 1990.

Townsend, C., *Mastering dBASE IV 1.1 Programming*, 2nd ed., Sybex, Inc., 1991.

Whitten, J.L., Bently, L.D., and Barlow, V.M., *Systems Analysis & Design Methods*, 2nd ed., Irwin, 1989.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Professor Shu Sheng Liao, Code AS/Lc Department of Administrative Sciences Naval Postgraduate School Monterey, CA 93943-5002	1
4.	Capt. Otis C. Ferrell, USN (Ret) Naval Postgraduate School P.O. Box 8626 Monterey, CA 93943	1
5.	LT Mark Kohlheim P.O. Box 454 Cool, CA 95614	1